

SVEUČILIŠTE U ZAGREBU
FILOZOFSKI FAKULTET
ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE
ZNANOSTI

Ak. god. 2014./2015.

Marko Požega

**Primjer informacijskog sustava u web 2.0 okruženju
(Projekt praćenja kretanja novčanica hrvatske kune u Republici
Hrvatskoj)**

Završni rad

Mentor: dr. sc. Kristina Kocijan, doc.

Zagreb, 2015.

Sadržaj:

1. Uvod	4
2. Osnovni pojmovi	5
3. Projekt „Praćenje kretanja novčanica hrvatske kune u Republici Hrvatskoj“	9
3.1. „Currency bill tracking“	9
3.2. Primjer postojećih projekata (web stranica/aplikacija)	9
3.2.1. „Where is George“	10
3.2.2. EuroBillTracker.....	11
3.2.3. Where's Willy?	12
3.2.4. Track Dollar	13
3.2.5. Canadian Money Tracker (CMT).....	13
3.2.6. EuroTracer.....	13
3.2.7. Where's Your Money?.....	13
3.2.8. Ostale web stranice.....	14
3.3. Hrvatska inicijativa.....	14
4. Projektiranje i realizacija web aplikacije (gdjemojakuna.hr)	17
4.1. On-line server i internetska domena	17
4.2. Informacijska shema i struktura web aplikacije	19
4.3. Organizacija i strukturiranje baze podataka	20
4.4. HTML struktura web stranice.....	23
4.5. Dizajn mrežne aplikacije	26
4.6. Implementacija skriptnih klijentskih jezika, otvorenog koda i API-a	29
4.7. Programiranje mrežne aplikacije	31
4.8. Optimizacija koda mrežne aplikacije	34
4.9. Pretvorba mobilne verzije mrežne aplikacije u mobilnu android aplikaciju	35

4.10.	Sigurnost i zaštita podataka.....	35
4.11.	Budućnost web aplikacije.....	37
5.	Zaključak.....	39
6.	Literatura	40

1. Uvod

U ovom završnom radu opisat ću proces nastajanja, projektiranja, dizajniranja i redizajniranja informacijskog sustava u modernom web 2.0 okruženju. Kao primjer jednog takvog informacijskog sustava odabrao sam mrežnu aplikaciju na mrežnoj adresi (www.gdjejemojakuna.hr) koja je rezultat zajedničkog projekta „Praćenja kretanja novčanica hrvatske kune u Republici Hrvatskoj“ u suradnji s mentoricom dr. sc. Kristinom Kocijan, doc. i kolegicom Enom Fuzul. O samom projektu, njegovoj misiji i viziji govorit ću u trećem poglavlju. Veći naglasak u ovome radu stavit ću na mrežne tehnologije, programske jezike, metode i tehničke ideje koje su korištene u izgradnji ovog sustava.

U drugome poglavlju rada objasniti ću osnovnu terminologiju koja je usko povezana s temom rada i čije je poznavanje nužno za daljnje razumijevanje rada, konkretnije programskih jezika i tehnologija korištenih u projektu. U trećem dijelu objasniti ću samu ideju projekta, misiju, viziju i svrhu provođenja ovakvoga projekta. Nadalje, usporedit ću ovaj projekt sa sličnim svjetskim projektima od kojih je došla motivacija i ideja za radom na istom projektu. U trećem dijelu rada dočarat ću ideju funkcionalnosti mrežne aplikacije, organizacijsku strukturu baze podataka sustava te tehnologija koje su bile potrebne za izgradnju jednog takvog sustava. Detaljnije ću objasniti sva tehnička rješenja i probleme na koje sam nailazio tijekom razvijanja sustava. Upravo zbog nastalih problema odlučio sam mijenjati organizacijsku strukturu aplikacije i razviti drugu verziju mrežne aplikacije.

U završnome dijelu rada iznijet ću zaključak temeljen na teorijskom i praktičnom pristupu izgradnje informacijskog sustava u mrežnom okruženju. Navest ću odrednice koje bi trebalo slijediti kako bi se što prije i kvalitetnije došlo do konačnog proizvoda i izbjegle neželjene komplikacije i redizajniranja sustava. Također, cilj ovog rada je naglasiti jednostavnost i prednosti izrade informacijskih sustava u mrežnom okruženju, opisati mogućnosti i ograničenja PHP, JavaScript, HTML, CSS, JQuery programskih jezika i jezika za označivanje u razvoju interaktivnih mrežnih aplikacija, naglasiti važnost informatičke etičnosti, važnost *open-source* projekata u nastanku drugih, naznačiti pojam i sverasteću ulogu velikih podataka u organizacijama bilo kojeg karaktera te potaknuti neke od čitatelja na mogući samostalni razvoj sličnih sustava.

2. Osnovni pojmovi

U ovome poglavlju objasniti ću osnovnu terminologiju koja je usko povezana s temom rada i čije je poznavanje nužno za daljnje razumijevanje rada, konkretnije programskih jezika i tehnologija korištenih u projektu.

Informacijski sustav

Informacijski sustav je sustav koji prikuplja, pohranjuje, čuva, obrađuje i isporučuje podatke svim članovima neke organizacije kojima su potrebni. Koristi se u računalnim znanostima, matematici, geografiji, sociologiji, sistemskoj teoriji, telekomunikacijama, menadžmentu i drugim znanostima (Radovan, 1991.). U današnje vrijeme rijetko koja organizacija može funkcionirati bez informacijskog sustava, jer informacijski sustav, temeljem tehnika koje se koriste u grani umjetne inteligencije, nastoji iz podataka izlučiti informacije koje poslovodstvo koristi za donošenje odluka. Stoga možemo reći da moderan informacijski sustav služi za potporu odlučivanju, kao što je u ovom projektu odrađivao funkciju pribavljanja podataka od strane korisnika, pohranjivanja u bazi podataka te izvođenje statističkih podataka na temelju istih.

Web 2.0 okruženje

Web 2.0 predstavlja WWW (eng. *World Wide Web*) stranice čije karakteristike omogućavaju suradnju i razmjenu sadržaja među korisnicima web usluga (Carnet, 2015.). Ovaj izraz prvi se put pojavio 1999. godine napretkom mrežne tehnologije i pojavljivanjem raznih društvenih trendova i načina razmjene informacije znanja poput blogova i foruma. Takav kontekst mrežnog sadržaja otvorio je mogućnost nastanku mnogih formi društvenih mreža, sustava za razmjenu multimedijalnog sadržaja, mrežnih aplikacija i slično. Okruženje kao takvo ostavlja „plodno tlo“ za razvoj informacijskih sustava i mrežnih aplikacija.

Baze podataka

Baza podataka može se definirati kao organizirana zbirka zapisa pohranjenih na računalu na sustavni način, gdje se svaki zapis za bolji povratak i razvrstavanje obično prepoznaje kao skup elemenata (činjenica) podataka. Sustav pohrane baze podataka je takav da omogućava

programu da mu se obrati prilikom odgovaranja na problem. Takav program koji se koristi za upravljanje i ispitivanje baze podataka naziva se sustavom za upravljanja bazom podataka (eng. *Data base management system*) (Radovan, 1993.). Baze podataka čine temelj svakog informacijskog sustava kao što je primjer i u našem slučaju. Za razvoj naše web aplikacije koristili smo MySQL bazu podataka smještenu na web serveru. Glavna svrha baze podataka u našem projektu je da se u nju pohranjuju svi podatci vezani za registracije novčanica, a uključuju podatke o: lokaciji registracije, vremenu registracije, korisničkim računima i ostalim podacima koje ću detaljnije objasniti u poglavlju 3.3.

SQL

SQL je kratica za *Structured Query Language*. SQL je strukturirani jezik za manipuliranje bazama podataka, a služi za pristup i manipuliranje velikim količinama podataka unutar baze podataka (W3Schools, 2015.). Ovaj jezik korišten je za manipulaciju bazom podataka naše web aplikacije koji se izvršavao kroz PHP programski kod.

HTML

HTML je kratica za *HyperText Markup Language*, odnosno prezentacijski jezik za izradu web stranica. Njime se oblikuje sadržaj i stvaraju se hiperveze hipertekst dokumenta. Temeljna zadaća HTML jezika je uputiti mrežni preglednik kako prikazati hipertekstualni dokument, nastojeći pritom da taj dokument izgleda jednako, bez obzira o kojem je pregledniku, računalu i operacijskom sustavu riječ. HTML nije programski jezik, njime ne možemo izvršiti nikakvu zadaću, on nam služi samo za opis naših hipertekstualnih dokumenata. HTML je jednostavan za upotrebu i lako se uči, zbog čega je opće prihvaćen i popularan (W3Schools, 2015.). Pomoću HTML jezika formirana je struktura i naše web stranice kreiranjem elemenata za unos i prezentaciju podataka poput: zaglavlja, footera, izbornika, upitnih formulara itd.

CSS

CSS je kratica od *Cascading Style Sheet*. Radi se o dokumentu u kojem se definira izgled HTML stranice, a njegova osnovna funkcionalnost je u činjenici da se skup pravila za prikaz primjenjuje nad svim elementima istog imena. Najčešće se radi o jednom dokumentu koji se primjenjuje nad cijelim mrežnim središtem (W3Schools, 2015.). Na našoj web stranici CSS

jezik za označivanje korišten je za definiranje stilističkih atributa HTML elemenata kako bi se postigao privlačan i intuitivan dizajn (izgled) stranice.

JavaScript

JavaScript je skriptni jezik koji se najviše koristi kod izrade mrežnih stranica, izvodi se s klijentske strane, odnosno na uređaju koji otvara mrežnu stranicu. Po definiciji može biti unutarnji i vanjski. Unutarnji je napisan unutar HTML dokumenta, dok je vanjski napisan u posebnom dokumentu koji se zatim poziva na sličan način kao i *css* datoteka. JavaScript je ujedno i najpopularniji skriptni jezik na svijetu (W3Schools, 2015.).

jQuery

jQuery je posebna vrsta *JavaScript* biblioteke, razvijene s namjenom da bude nadogradnja osnovnog *JavaScript*-a. *jQuery* pojednostavljuje njegovu sintaksu i omogućava bolju interakciju između *JavaScript*-a i drugih programskih jezika namijenjenih razvoju mrežnih aplikacija. (W3Schools, 2015.). Zajedno s osnovnom *JavaScript*om, ovaj skriptni jezik u našem projektu korišten je za upravljanje HTML elementima te njihovim stilističkim svojstvima kako bi se postiglo bolje korisničko iskustvo. Nadalje korišteni su za upravljanje API sučeljima, dinamičko kontroliranje HTML obrazaca itd.

PHP

PHP je skriptni serverski programski jezik sličan *JavaScript*-u koji služi prvenstveno za programiranje dinamičnih web stranica, ali se za razliku od *JavaScript*-a izvodi na serveru i stoga nije ovisan o tome što podržava uređaj na kojem se mrežna stranica pregledava. Orijentira se po *C* i *Pearl* sintaksi. PHP se ističe širokom podrškom raznih baza podataka i internetskih protokola (W3Schools, 2015.).

Open Source

Otvoreni kod (en. *Open source*) je način objavljivanja računalnih aplikacija koji omogućava onima koji ih dobiju da ih mijenjanju i dalje dijele. Računalne aplikacije objavljene kao *otvoreni kod* ne moraju biti besplatne, iako često jesu (Wikipedia, 2015.). U našoj web aplikaciji koristila su se gotova programska rješenja otvorenog koda uglavnom vezana za skriptni *jQuery* kod. Neki od gotovih programskih rješenja korištenih u projektu su: optičko

prepoznavanje znakova, iscrtavanje ruta unutar *Google Maps* karti, modificirani prozori obavijesti itd.

API

API je kratica za *Aplikacijsko Programsko Sučelje* (eng. *Application Programing Interface*). API je skup određenih pravila i specifikacija koje programeri slijede da bi se mogli služiti uslugama ili resursima operacijskog sustava ili nekog drugog složenog programa kao standardne biblioteke rutina, struktura podataka, objekata i protokola (Begušić, 2011.).

Big data (Veliki podatci)

Termin *Big data* ili *Veliki podatci* osmislili su znanstvenici iz područja geonomije i astronomije jer se u tim područjima količina podataka mjeri u ekstremnim količinama, pa se tako *Big data* ili *Veliki podatci* najjednostavnije mogu definirati kao svi podatci kojima nije moguće upravljati, niti ih analizirati pomoću standardnih alata i tehnika analize podataka (Kocijan, 2015.). Ovaj termin smatram relevantnim za projekt zbog toga što se „širenjem“ baze podataka i nagomilavanjem podataka o kretanju novčanica otvaraju vrata za otkrivanjem nekih uzoraka i zaključaka koji primarno nisu definirani na početku projekta.

3. Projekt „Praćenje kretanja novčanica hrvatske kune u Republici Hrvatskoj“

Prije nego što opišem navedeni projekt koji je započet u Republici Hrvatskoj, objasniti ću važne termine i navesti primjere drugih projekata iz drugih država. U zadnjih petnaestak godina, potaknuta pojavom web-a 2.0, pojavila se zanimljiva ideja o praćenju kretanja novčanica unutar i van države putem interneta. Pojavom projekata na internetu temeljenih na navedenoj ideji skovao se termin pod nazivom *Currency bill tracking*, što bi u grubom prijevodu značilo „Praćenje valuta (novčanica)“. No zbog okruženja u kojem se praćenje svih ovih godina obavlja, osobno bi preveo termin kao „Praćenje kretanja novčanica valute putem interneta“.

3.1. „Currency bill tracking“

Praćenje valuta putem interneta je proces (obično olakšan kreiranjem mrežne stranice koja služi u tu svrhu) praćenja kretanja novčanica, slično kako i ornitolozi prate migracije ptica. Mrežna stranica za praćenje kretanja novčanica može pratiti valutu među korisnicima te mrežne stranice. Korisnik može „registrirati“ lokaciju novčanice tako da unese podatke o serijskom broju novčanice te njenu lokaciju, a ako je netko već „registrirao“ novčanicu, onda se „trag“ novčanice može na neki određeni način reprezentirati.

Iz takvih podatka mogu se izvesti zaključci o prijašnjem putu izraženom u nekoj mjernoj jedinici, npr. brzina kojom se ta novčanica kretala te mnogi drugi statistički podatci koji se zasnivaju na puno većem reprezentativnom uzorku. Glavni cilj svih projekata zasnovanih na takvoj tematici je potaknuti korisnika sustava, a i što veći broj ljudi da se svaka novčanica prije potrošnje obilježi odnosno „registrira“ – ostavi iza sebe trag u obliku podataka. Na takav način, u teoriji, bilo bi moguće pratiti kretanje svake individualne novčanice jedne države, te na taj način imati potpun uvid u putovanje i životni vijek novčanice (Wikipedia, 2015).

3.2. Primjer postojećih projekata (web stranica/aplikacija)

U narednom potpoglavlju navest ću postojeće instance mrežne aplikacije za praćenje kretanja novčanica. Podrobnije ću objasniti funkcionalnosti najvećih takvih stranica i aplikacija, neke „američke“ i neke „europske“ instance. Njihova usporedba dana je u Grafu 1.

3.2.1. „Where is George“

Prva takva ideja realizirala se u obliku mrežne stranice koja se zasnivala na unosu podataka od strane korisnika stranice o lokaciji novčanice jednog američkog dolara (kasnije i novčanica ostalih nominalnih vrijednosti). Kreator projekta Hank Eskin odabrao je kreativan naziv za svoju mrežnu stranicu - „Where's George?“ koja se nalazi na mrežnoj adresi <http://www.wheresgeorge.com>.

Ova stranica je od velike važnosti ne samo zbog toga što je njeno začecje poteklo razvoj mnogih drugih mrežnih stranica za praćenje novčanica već je razvilo novi koncept praćenja stvari putem interneta poput praćenja starih korištenih knjiga. „Where's George“ ispočetka nije imao konkretno definiran cilj, već je stranica isključivo služila za evidenciju kretanja novčanica. Međutim s vremenom i povećanom frekvencijom korištenja stranice, mnogi statistički zaključci su počeli isplivavati na površinu. O tome govori podatak da su statistički podaci generirani od strane ove stranice korišteni u istraživanju.

Najznačajnija primjena dobivenih podataka iskorištena je 2006. godine u istraživačkom radu za predviđanje brzine širenja gripe u SAD-u za koja je korišten podatak o prosječnoj brzini kretanja novčanice američkog dolara (ScienceBlog.com, 2006.). U 2009. godini podatci sa iste stranice korišteni su u sličnom istraživanju za predviđanje širenja epidemije svinje gripe (Donald, 2009.).

Stranica je svoju popularnost u ranim godinama 21. stoljeća iskoristila kako bi stvorila malu natjecateljsku igru na način da je stranica rangirala registrirane korisnike stranica kao natjecatelje. Naime, smišljen je poseban težinski koeficijent (bodovanje) koje se zasnivalo na aktivnosti, broju unesenih novčanica te broju „hitova“ odnosno broju „pogođenih novčanica“. Novonastali termin „hit“ u ovom kontekstu odnosi se na registriranje već registrirane novčanice u sustavu, čime se postiže „pogodak“- ono čemu taj sustav i teži. „Natjecatelji“ od svojih rang mjesta nemaju nikakve materijalne beneficije osim lijepog priznanja i ispisivanja imena u vrhu stranicu.

Logaritamska formula za izračun rang mjesta korisnika („The George score“):

$$100 \times [\sqrt{\ln(\text{broj unešenih novčanica})} + \ln(\text{broj pogodaka} + 1)] \times [1 - (\text{broj dana neaktivnosti}/100)]$$

Ova logaritamska formula generira rezultat na temelju broja registriranih novčanica, broja postignutih pogodaka te vremena neaktivnosti korisnika na stranici. Što više novčanica

korisnik registrira i što više pogodaka ostvari korisnik će imati veći rezultat (Where's George? L.L.C, 2015.).

Međutim u formulu ulazi varijabla aktivnosti odnosno neaktivnosti korisnika koja korisnika tjera da frekventno registrira novčanice. Tako da se korisniku unatoč velikom broju registriranih novčanica i broja pogođenih novčanica oduzimaju bodovi za svaki dan neaktivnosti. Čime se korisnika potiče na aktivnost ako želi zadržati određeno mjesto na rang listi.

No, postavlja se pitanje kako se financirati iz ovakve vrste projekta? *Where's George* je svoj problem riješio dobrim, starim reklamiranjem te prodajom suvenira. Stranica je preplavljena raznim reklamama što nekim stalnim posjetiteljima stvara veliku frustraciju. No i za to postoji rješenje, jer stranica nudi *ad-free* korisničko iskustvo, no uz nadoplatu od 8 dolara mjesečno, uz neke dodatne funkcije poput detaljnijeg pregleda baze podataka.

Nakon 16 godina aktivnosti stranica je skupila impresivan uzorak registriranih novčanica. Tako je na primjer do dana 21.8.2015 (Where's George? L.L.C, 2015.):

- Uneseno je sve ukupno 254,032.150 novčanica;
- Ukupna vrijednost unesenih novčanica izražena u valuti američkog dolara iznosi 1,365,930.870\$;
- Ukupna vrijednost izražena u postotku od cjelokupne količine novca u optjecaju iznosi 0.118%;
- Prosječni postotak *pogodaka* u odnosu na broj registriranih novčanica iznosi 11.4 %;
- Rekord stranice je novčanica od jednog dolara koja je postigla čak 15 *pogodaka*.

Bez dvojbe može se reći da je stranica postigla neskroman uspjeh te uspjela privući korisnike. No mnogo veći uspjeh stranica je postigla tim što je služila kao primjer i ideja za daljnje nastale projekte i mrežne stranice, kao što je bila najvažniji faktor za pokretanje jednog takvog projekta u Republici Hrvatskoj.

3.2.2. EuroBillTracker

Samo tri godine nakon nastanka *Where is George* stranice, po njezinom primjeru pokrenut je novi projekt koji se ovoga puta zasnivao na praćenju valute eura. Kao i kod prvog projekta, cilj je bio i ostao sakupiti što više *registriranih* i *pogođenih* novčanica kako bi se moglo izlučiti što više informacija vezanih za rasprostranjenost novčanica po zemljama članicama

eurozone (Wikipedia, 2015). EuroBillTracker se deklarira kao samostalna neprofitabilna volonterska udruga koja želi ostvariti gore navedeni cilj.

Za razliku od „*Where is George*“, ova mrežna stranica je u potpunosti besplatna za korisnike, ne financira se čak niti iz reklamnih sadržaja, što pruža dosta mirnije iskustvo korištenja stranice. Nadalje, organizacija nije u interakciji s Europskom unijom niti Europskom centralnom bankom.

Ostale karakteristike su manje više poprimljene od „*Where is George*“ mrežne stranice. Tako se za registraciju novčanica potrebno registrirati pod određenim korisničkim imenom, za svakog korisnika se izvodi individualna statistika na temelju koje se radi već spomenuto rangiranje koje funkcionira na sličnom principu. Veliki naglasak se stavlja i na već spomenute „pogotke“.

Niti kod ove stranice ne možemo zanemariti neke statističke podatke koje je stranica postigla, tako do dana 24.8.2015 broji (Eurobilltracker.com, 2015.):

- 194.818 korisnika;
- 147,777.154 unesenih novčanica ukupne nominalne vrijednosti od 2,763,235.775 €;
- 838.920 „interesantnih“ pogodaka. Pod interesantnim pogocima smatraju se pogoci onih novčanica koje su prošle kroz najmanje dvije europske države.

3.2.3. Where's Willy?

Po potpuno istom uzorku kao i „*Where is George*“, 2001. godine ponovno pod rukom Hanka Eskina nastaje mrežna stranica za praćenje kretanja kanadskog dolara pod nazivom „*Where's Willy?*“. Stranica radi na gotovo istom principu kao i prvo nastala mrežna stranica. Bazira se na unosu korisnika koji se za korištenje mrežne stranice mora registrirati. Stranica prati pogotke i rangira korisnike po algoritmu definiranom u 3.2.1. potpoglavlju, no u ovom slučaju nosi naziv *Willys index* (Wikipedia, 2015.). Iako rađena po istom modelu i od strane istog autora, ova mrežna stranica nije doživjela očekivani uspjeh kao što je bio slučaj s američkom inačicom stranice.

Do dana 24.8.2015 stranica broji (Where's George? L.L.C, 2015.):

- 4,483.626 registriranih novčanica u ukupnoj vrijednosti od 67,457.150 \$;
- Samo 14 postignutih pogodaka.

3.2.4. Track Dollar

TrackDollar.com je stranica posvećena praćenju novčanica američkog dolara diljem svijeta. Radi na standardnom principu unosa serijskog broja i informacije o lokaciji za svaku novčanicu koja se registrira. Na temelju tih informacija stranica daje informacije o praćenju te statistike i rangiranja. Stranicu je osnovao neprofitni tim volontera 20.09.2008. Dosad je uneseno 771.828 novčanica, u ukupnoj vrijednosti od \$3,542.103 (Trackdollar.com, 2015.).

3.2.5. Canadian Money Tracker (CMT)

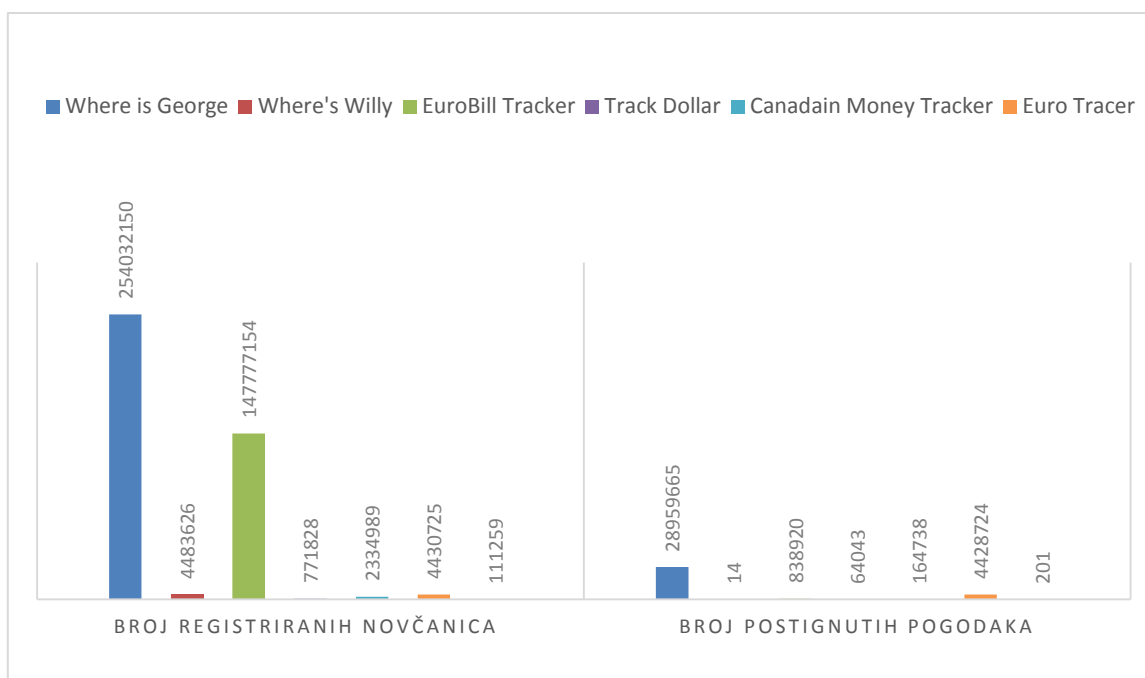
Canadian Money Tracker je mrežna stranica koja prati novčanice kanadskog dolara. Stranica na temelju korisničkih računa pruža korisniku podatke gdje je i u čijem posjedu korisnikova novčanicu bila. Radi na istom principu kao i TrackDollar.com. Stranica je osnovana 1999. god., a dosad je uneseno 2,334.989 novčanica u ukupnoj vrijednosti od \$36,850.738 (Cdn-money.cd, 2015.).

3.2.6. EuroTracer

EuroTracer je mrežna stranica koja prati kretanje novčanica euro valute diljem Europe i van nje. Stranica pruža korisniku uvid u kretanje novčanica, statistiku i rang liste. Ova stranica ima mogućnost praćenja i kovanica euro valute. Stranica je osnovana 2002. god., a do danas je uneseno 4,414.518 novčanica u vrijednosti od 109,641,500 € te 505,665 kovanica u vrijednosti od 261,592.18 € (Eurotracer.net, 2015.).

3.2.7. Where's Your Money?

Wheresyourmoney.com je stranica osnovana 2009. god. pod nazivom *Find Lizzy*. U početku je služila kao stranica za praćenje britanske funte, a od 2012. god. nakon redizajna i preimenovanja stranice započinju s projektom praćenja američkog dolara, britanske funte i euro valute (Findlizzy.co.uk, 2015.).



Graf 1: Usporedba količine prikupljenih podataka s različitih web stranica

3.2.8. Ostale web stranice

Ostale manje uspješne i poznate stranice za praćenje kretanja novčanica (Wikipedia, 2015):

- *DoshTracker* - Osnovana je 2001. godine i prati kretanje britanske funte;
- *CashFollow* - Švicarska stranica osnovana 2001. godine koja se bavi praćenjem švicarskog franka;
- *Sek-tracker* - Stranica osnovana 2004. godine, a svrha joj je praćenje kretanja švedske krune;
- *The Money Tracker* - Osnovana 2006 i prati australski dolar;
- *Wheres my moola* - Prati valutu južnoafričkog randa, osnovana je 2011;
- *Wheres Renminbi* - Osnovana 2006. godine, prati kineski yuan;
- *Wheres NT* - Prati tajvanski dolar;
- *Ontanebito* - Prati kretanje meksičkog pesosa;
- *Cupura.Ru* - Prati kretanje ruskog rublja.

3.3. Hrvatska inicijativa

Po primjeru mrežne stranice *Where is George* došli smo do ideje pokretanja projekta sličnog karaktera u Republici Hrvatskoj. Naravno, zadaća našeg projekta bila bi praćenje kretanja novčanica hrvatske kune unutar granica Republike Hrvatske i nekih susjednih zemalja.

Prvobitna ideja zasnivala se na kreiranju stolne (*eng. desktop*) mrežne stranice u kojoj bi se od korisnika tražio standardan unos serijskog broja novčanice i podatak o lokaciji tekstualnog formata koji se u početku odnosio samo na grad/općinu/mjesto unosa. Međutim, već u ranoj fazi rješenja naišli smo na neke probleme i pitanja koja su trebala biti riješena. Glavni problem bio je problem legitimnosti i vjerodostojnosti podataka bez kojih niti jedno istraživanje ili projekt istraživačkog karaktera nema čvrst oslonac.

Najvažniji problem se odnosi na verifikaciju unesenog serijskog broja. Budući da nismo imali mogućnost uvida u popis svih izdanih novčanica hrvatske kune od samoga početka tiskanja, nismo mogli verificirati koja je novčanica „ispravna“, a koja se može registrirati kao *spam*. Taj problem u ranijoj fazi riješen je relativno plitko, na način generiranja regularnog izraza koji će kontrolirati unos u obliku da se na početku nalaze veliko slovo A ili veliko slovo B iza kojih slijedi sedmeroznamenasti broj nakon kojih slijede jedno od velikih slova hrvatske abecede bez dijakritičkih znakova. No to rješenje nije u potpunosti legitimno što je dovelo do originalne ideje koja do sada nije primijenjena na niti jednu od postojećih stranica za praćenje novčanica.

Radi se inicijativi registriranja novčanica putem mobilnog uređaja putem fotografiranja novčanice i *OCR* tehnologije. Takvo rješenje zahtijevalo je kreiranje mrežne stranice prilagođene mobilnim uređajima te treniranje i implementiranje *OCR* tehnologije u prepoznavanju znakova s novčanice. Tim činom se stvara potpuna legitimnost podatka o serijskom broju novčanice. Inzistiranjem na mobilnom rješenju pojavilo se i rješenje drugog problema legitimnosti, a to je podatak o lokaciji. Naravno da bi se u stolnoj verziji mrežna stranica ručnim unosom od strane korisnika mogla lažirati lokacija, što bi kvarilo legitimnost podataka i neke buduće zaključke temeljene na njima. Stoga se u mobilnoj verziji stranice korisniku ne omogućava ručni unos lokacije već je primoran podijeliti svoju lokaciju putem GPS uređaja smještenog u samom mobilnom uređaju. S tim se dobivaju vrlo precizni podatci o lokaciji koji se ne mogu lažirati. Kako bi se ovo mrežno rješenje još više približilo mobilnim uređajima, napravljena je i android aplikacija.

Ostatak mrežne stranice koncipiran je na sličan način kao kod navedenih stranica, a sastoji se od mogućnosti pretraživanja baze podatka, stranice koja dinamički generira i prezentira statističke podatke izvedene iz nastale baze podataka, te stranice za administraciju korisničkog računa. U početnom stadiju projekta osmišljen je naziv „Info-kune“ no u kasnijem razvoju odlučili smo se za naziv „Gdje je moja kuna“ zbog toga što je početni naziv

više asocirao na pružanje informacije o hrvatskoj kuni kao pojmu dok drugi naziv implicira na njenu lokaciju što je i temeljna tema projekta. Na temelju odabranog naziva osmišljen je i logotip koji je prikazan na Slici 1.

Kao cilj projekta mogla bi se definirati teorijska pretpostavka da kada bi svaki državljanin Republike Hrvatske svakog dana ispraznio svoj novčanik i registrirao novčanice koje posjeduje, kreirala stvarna slika kretanja i kolanja novčanica Hrvatske kune unutar i van Republike Hrvatske. Jedan takav „otisak“ bi zasigurno bio od dubokog informativnog, a možda i istraživačkog potencijala. No realizacija takve ideje zahtijeva financiranje i marketinške mjere koje u ovom stadiju projekta nisu moguće. Zato se kao početan cilj pokušava osvijestiti uži krug stanovništva, prvenstveno akademske građanine na korištenje web aplikacije kako bi možda jednoga dana ostvarila poticaj od neke interesne skupine.



Slika 1: Logo projekta i web stranice

4. Projektiranje i realizacija web aplikacije (gdjejemojakuna.hr)

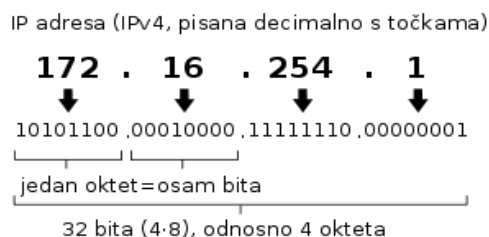
U ovom poglavlju detaljno ću opisati proces kreiranja i optimizacije web aplikacije na web adresi www.gdjejemojakuna.hr. U svakom od narednih potpoglavlja objasniti ću metode, tehnička i programska rješenja korištena u dobivanju konačnog dizajnerskog i programskog proizvoda.

4.1. On-line server i internetska domena

Naravno, kako se radi o mrežnoj aplikaciji, odnosno informacijskom sustavu u web okruženju, cijeli funkcionalni proces mora se odvijati na internetu. Kako bi si osigurali mjesto na internetu svoje programsko rješenje potrebno je pohraniti na jednom od javnih ili privatnih servera koristeći validnu internetsku domenu.

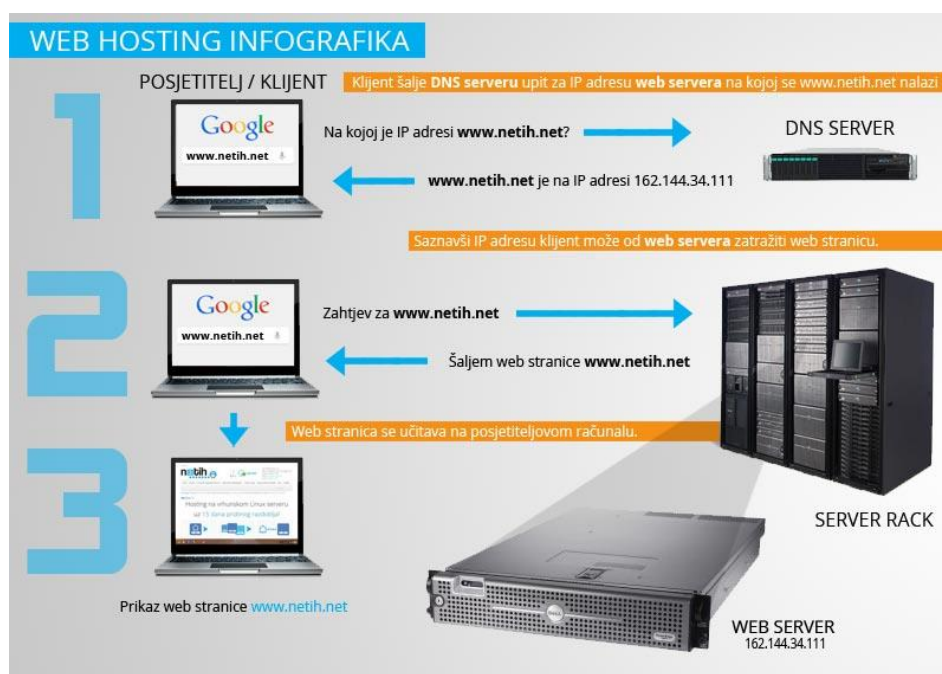
Svaki je mrežni poslužitelj (eng. *Web server*) jednostavno definirano računalo koje je mrežnim putem povezano s „većim“ računalima koji zajedno tvore internet. Web server kao i svako računalo ima svoj podatkovni prostor koji se može koristiti u svrhu pohranjivanja i učitavanja web stranica koje korisnici mogu javno pregledavati sa svojih klijentskih računala (Preston, 2015.).

Internetsku domenu možemo definirati kao ime ili neki naziv pomoću kojeg se neko računalo (u našem slučaju spomenuti server i web stranica) identificira na internetu. Stvarni podatak o lokaciji računala u nekom mrežnom sustavu naziva se IP adresa (eng. *Internet protocol adress*) koja je u svom formatu nepraktična za pamćenje i korištenje (Slika 3) stoga je „zakodirana“ u prirodnom jeziku kojeg DNS server (*Domain Name System*) prevodi u IP adresu. (Preston, 2015.).



Slika 3: Primjer strukture IP adrese¹

Svaki web server ima IP adresu. Kada se posjećuje neka web stranica klijentsko računalo ustvari traži IP adresu servera na kojoj su web stranice smještene. Kada dobije IP adresu računalo zna od kojeg web servera mora tražiti web stranice. Tek onda se web stranice mogu prikazati na klijentskom računalu. Ovaj proces ilustriran je Slici 4.



Slika 4: Shema protokola dohvaćanja web stranice s web servera²

Zato jer je kupnja, konfiguracija i održavanje servera skup i kompleksan proces, većina današnjih klijentskih potreba za objavljivanjem neke web stranice odvija se putem web hosting usluga. To je usluga u kojoj klijent plaća podatkovni prostor i domenu za korištenje u određenom vremenskom periodu na specifičnom serveru kojeg posjeduju davatelji usluge. Cijena hosting usluge određena je posebnim faktorima poput: količine podatkovnog prostora,

¹ Slika preuzeta sa: <http://www.netih.net/info/sto-je-hosting.html>

² Slika preuzeta sa: <http://www.netih.net/info/sto-je-hosting.html>

vrste servera (PHP/Apache/Linux), podrškom za MySQL baze podataka, broja poddomena, količina maksimalnog dnevnog prometa, postotka opterećenja glavnog procesora servera itd.

U našem slučaju koristimo *Apache* server s *PHP* i *MySQL* podrškom.

4.2. Informacijska shema i struktura web aplikacije

Prije samog početka izgradnje web stranice potrebno je odrediti od kojih se komponenti mora sastojati. Svaki informacijski sustav treba sadržavati komponentu u koju informacije ulaze, komponentu u kojoj se informacije obrađuju, sortiraju, organiziraju te komponentu u kojoj se informacije prezentiraju.

Naša web aplikacija sadržava:

- index.php – početna stranica koja sadržava kratak opis projekta i najnovije statistike;
- unos.php – stranica na kojoj će se vršiti registracija novčanica;
- unos_bp.php – stranica koja će se izvršavati nakon registriranja novčanice, a u kojoj će se vršiti obrada i verifikacija podataka, unos podataka u bazu podataka te ažuriranje postojećih podataka ovisno o unesenom podatku;
- pretraga.php – stranica koja će omogućavati pretraživanje baze podataka po serijskom broju ili mjestu registracije novčanice. Kao rezultat pretrage dobit će se detaljne statistike vezane za upit, ovisno o tome nalazi li se ili ne traženi serijski broj ili mjesto u bazi podataka;
- statistika.php – stranica koja je zadužena za prezentaciju velike količine podataka. Na ovoj stranici grafički i tablično se prezentiraju svi statistički podatci iz baze podataka kategorizirani po nominalnim vrijednostima novčanica te se generiraju dinamične rang liste i grafovi;
- projekt.php – stranica informativnog karaktera koja pruža detaljan opis projekta kao i njegov napredak i budući plan;
- korisnicki_racun.php – komponenta koja omogućava korisniku web stranice uvid u dosad registrirane novčanice, podešavanje dodatnih opcija poput mogućnosti kontakta putem e-mail adrese u slučaju „pogotka“ itd.

4.3. Organizacija i strukturiranje baze podataka

Prije nego što započnemo s izgradnjom same web stranice potrebno je definirati neke logističke okvire po kojima će stranica funkcionirati. Kao komponenta bez koje niti jedan informacijskih sustav ne može funkcionirati, baze podataka zahtijevaju detaljno definiranu strukturu, promišljeni organizacijski i relacijski ustroj te provedenu normalizaciju.

Normalizacija baze podataka odnosi se na proces u kojem se iz modela baze podataka pokušava ukloniti redundantnost, odnosno ponavljanje podataka (Radovan, 1993).

No prije same organizacije baze podataka, kreiranja tablica i relacijskih odnosa među njima, potrebno je odrediti koji podatci ulaze u sustav te na kojih ih način rasporediti što uvelike određuje način i oblik na koji ćemo zapisane podatke prezentirati. U ovom procesu bitno je razmišljati unaprijed jer je prebacivanje baze podataka iz jednog oblika u drugi zbog neke novonastale ideje ili druge naravi vrlo skup i zahtjevan posao naročito ako se radi o velikom i kompleksnom informacijskom sustavu.

Sličan problem pojavio se nakon izgradnje prve verzije naše aplikacije. Zbog nedovoljno dobre organizacije tablica i podataka unutar baze podataka nismo mogli provesti jednu od ideja nastalih nakon završetka prve verzije web stranice. Zbog toga smo odlučili promijeniti ustroj cjelokupne baze podataka kako bi se podatci, u slučaju novih ideja, mogli uspješno iskoristiti (primjer je dan u Tablici 1).

Smjernice na koje moramo obratiti pažnju pri kreiranju baze podatka, koje su korištene i u izgradnji naše baze podataka:

- Definiranje tipova podataka radi štednje prostorne memorije;
- Kodiranje teksta – ukoliko nije unaprijed definiran način na koji će se u bazu podataka kodirati neki tekstualni unos, postoji mogućnosti da se neki znakovi koji su nam možda vrijedni u zapisu neće zapisati u bazu podataka. Pri kodiranju također moramo obratiti pažnju na štednju prostora te moramo odabrati optimalno kodiranje za naše potrebe;
- Verifikacija unosa i obveza unosa – pri definiranju stupaca neke tablice možemo definirati je li polje obvezno popuniti pri unosu reda u bazu podataka. Također moramo paziti da u bazu podataka ne unosimo nelegitimne podatke. To možemo napraviti verifikacijom kroz programski jezik prije unosa u bazu podataka ili postavljanjem uvjeta za unos polja u bazi podataka. Verifikacija unosa se također

može postići uređivanjem relacijskog modela unutar baze podataka. U tom procesu određuju se relacije između polja tablica korištenjem sporednog ključa (Radovan, 1993);

- Normalizacija baze podataka - načelni je cilj normalizacije da baza podataka bude oblikovana tako da se svaki podatak upisuje samo jednom ili samo na jednom mjestu, još jednom u svrhu štednje prostora;
- Postojanje specifičnog identifikatora za svaki unos (red) u tablici - Definiranje primarnog ključa.

Naziv tablice: Novčanica_A000000A

Stupac	Tip podatka	Null	Zadana vrijednost
ID	int(11)	No	
Apoen	char(5)	Yes	NULL
DatumUnosa	date	Yes	NULL
DatumUnosaMod	char(10)	Yes	NULL
VrijemeUnosa	time	Yes	NULL
DrzavaUnosa	char(30)	Yes	NULL
MjestoUnosa	char(30)	Yes	NULL
AdresaUnosa	char(30)	Yes	NULL
GeoDrzavaUnosa	char(30)	Yes	NULL
GeoMjestoUnosa	char(30)	Yes	NULL
GeoAdresaUnosa	char(30)	Yes	NULL
BrojKilometara	float	Yes	NULL
Brzina	float	Yes	NULL
Osoba	char(30)	No	
Biljeska	tinytext	Yes	NULL
NacinUnosa	char(10)	Yes	NULL

Tablica 1: Primjer strukture jedne od tablica koja bilježi podatke o registraciji novčanice

4.4. HTML struktura web stranice

Kako bi naša, a i bilo koja web stranica bila funkcionalna potrebno je izgraditi njenu konstrukciju odnosno elemente preko kojih će informacije i podaci biti prezentirani. Pomoću HTML jezika za označivanje radimo upravo to, određujemo „kosti“ i strukturu web stranice.

HTML sintaksa sastoji se od oznaka elementa, atributa oznake elementa te njegove vrijednosti i vrijednosti oznake elementa. Oznake definiraju element dok se atributima definiraju njegova svojstva. HTML5 verzija je najnovija verzija ovoga jezika za označivanje i nju smo koristili u našoj web aplikaciji.

Za našu web stranicu korišteni su slijedeći elementi:

- `<div>` - označava prostor koji jedan element obuhvaća i izražava se u pikselima. Ovaj element jedan je od najvažnijih elemenata koji stranici daje dizajnersku strukturu (npr. Kodnik blok 1).

```
<div class="back" style="background: rgb(173,150,133);">
```

Kodni blok 1: Primjer korištenja `<div>` oznake na našoj stranici

- `` - element koji omogućava prezentaciju slikovnog sadržaja na web stranici (npr. Kodnik blok 2).

```

```

Kodni blok 2: Primjer korištenja `` oznake na našoj stranici

- `<video>`- HTML5 element koji omogućava reprodukciju video sadržaja na web stranici, u našem slučaju radi se o *streamingu* sadržaja stražnje kamere mobilnog uređaja.

Uz oznaku `<video>` u velikom broju slučajeva dolazi i oznaka `<canvas>` koja djeluje kao digitalno platno na koje možemo ispisivati slikovni sadržaj, u našem slučaju zamrznuta sličica video zapisa koja se kasnije analizira algoritmom za optičko prepoznavanje teksta (npr. Kodni blok 3).


```
<video id='video_zapis' autoplay > </video>
<canvas style="display:none" id='c1'> </canvas>
```

Kodni blok 3: Primjer korištenja `<video>` i `<canvas>` oznake na našoj stranici za dohvaćanje video zapisa sa stražnje kamere mobilnog uređaja

- `<form>` - element od najvećeg značenja kada se govori u kontekstu informacijskog sustava koji služi za sastavljanje upitnika iz kojih od korisnika crpimo neke nama bitne informacije. Važni atributi unutar `<form>` oznake su *method* i *action* kojima se definira gdje će se dobiveni podatci poslati te kojom metodom (*POST* ili *GET*) (npr. Kodni blok 4).
- `<input>` - elementi koji se definiraju unutar `<form>` elemenata a označavaju vrstu unosa podataka ili informacije od strane korisnika. Vrsta unosa definira se atributom *type*, a neki od njih su: Textbox, Radio button, Checklist, Submit button itd. (npr. Kodni blok 4).

```
<form action='prijava_obrada.php' method="post">
  <input onchange='provjera_prijava()' type="text" id='Email_login'
name="Email_login" placeholder="Unesite email" style="width:200px;" maxlength=20>
</br> </br>

  <input onchange='provjera_prijava()' type="password" id='Password_login'
name="Password_login" placeholder="Unesite odgovarajuću lozinku"
style="width:200px;" maxlength=20> </br> </br>

  <input class='button' type="submit" id='prijava_se' value="Prijava se" disabled/>

</form>
```

Kodni blok 4: Primjer kombiniranog korištenja `<form>` i `<input>` oznaka za prijavu korisničkim računom na našoj stranici

- `<table>` - vrlo važan element kada se govori u kontekstu prezentacije podataka. Ovim elementom moguće je kreirati tablicu bilo kojih dimenzija i atributa kako podatci bili što prirodniji za razumijevanje od strane korisnika. U domeni `<table>` oznake koriste

se dvije oznake za označavanje redova za tablice <tr>, te <td> za označavanje stupaca tablice (npr. Kodni blok 5).

```
<table border='2' style='color:black; border-color: rgb(80,82,112); border-radius:
10px;'>
    <tr>
        <td style='border-top-left-radius: px;'> Ukupan broj unosa </td>
        <td class='poravnato'> 1346 </td>
    </tr>
    ...

```

Kodni blok 5: Primjer korištenja <table>, <td> i <tr> oznaka na našoj stranici pri kreiranju tablica za prikaz statističkih podataka

- , , - elementi kojima se definiraju liste koje mogu biti numerirane ili nenumerirane (npr. Kodni blok 6).

```
<ul >

    <li> <a href="index.php">Naslovnica</a> </li>
    <li> <a href="#">Unos / Pretraga </a>

        <ul>
            <li><a href="unos.php">Unos</a></li>
            <li><a href="pretraga.php">Pretraga</a></li>
        </ul>

    </li>
    <li> <a href="statistika.php">Statistike</a> </li>
    <li> <a href="projekt.php">O projektu</a> </li>
    <li> <a href="novcanice.php">O novčanicama</a> </li>
    <li> <a href="korisnicki_racun.php">Korisnički račun</a> </li>

</ul>

```

Kodni blok 6: Primjer korištenja i oznaka na našoj stranici pri kreiranju navigacijskog izbornika

- `<p>`, `<h>`, ``, ``, `<e>`, `<q>`, `<hr>`, `</br>`, `<a>` - elementi zaduženi za segmentaciju, dekoraciju i povezivanje tekstualnog sadržaja unutar web stranice (npr. Kodni blok 7).

```

<p>

    Ideja za izradu ove web stranice i ovog projekta temeljena je na već postojećoj web
    stranici
    <a style="color:black;"
    href="https://www.wheresgeorge.com/"><e>www.whereisgoerge.com</e></a>.

    </br>

    Navedena stranica djeluje kao informacijski sustav za prikupljanje velike količine podatka
    (<b>Big data</b>) te automatsku i dinamičnu analizu istih.

    ...

```

Kodni blok 7: Primjer korištenja elemenata za ukrašavanje i segmentaciju teksta na našoj stranici

Dodavanje atributa *Style* unutar početnog dijela elementa moguće je definirati vizualne parametre napisane u *CSS* jeziku za označivanje koji je u našem slučaju pisan u odvojenoj datoteci i nije pisan unutar html stranice. Kako bi se iz zasebne *CSS* datoteke moglo opozvati na neki specifičan element unutar *HTML* datoteke koriste se *selektori*. Odnosno za svaki element u *HTML* dokumentu potrebno je dodati atribut *id* ili po potrebi atribut *class* kojima dodjeljujemo proizvoljna imena na kojima ćemo se opozivati unutar *CSS* datoteke (npr. Kodni blok 8).

4.5. Dizajn mrežne aplikacije

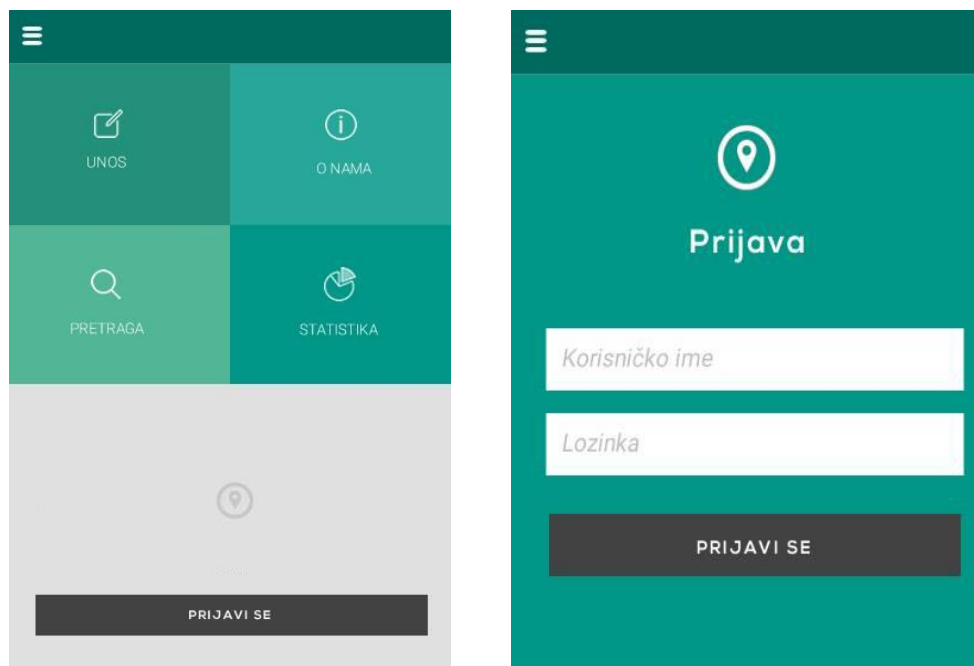
Dizajn naše stranice izrađen je da prati načela jednostavnog, skladnog i modernog dizajna. Dizajn se bazira na nijansama zelene boje koje asociraju na primarne funkcije aplikacije (praćenje novca), a kao kontrast korištena je komplementarna, crvena boja. Stavljen je naglasak na ikone i ostale potencijalne grafičke elemente kako bi se lakše, jednostavnije i brže izkomunicirala funkcija i značenje elemenata. U skladu s trendovima, dizajn je plošan (bez

prijelaza i nijansi) čime se postiže moderan izgled, stavlja naglasak na sadržaj i smanjuje količina dizajnerskih elemenata koji odvlače pažnju (Slika 5). Prilikom izrade korišten je „*Bootstrap*“, CSS razvojni okvir (eng. *Framework*) pomoću kojeg je olakšana izrada HTML i CSS mreže (eng. *Grid*) kako bi se jednostavno postigao konzistentan dizajn kako u mobilnoj tako i u stolnoj verziji (Slika 6).

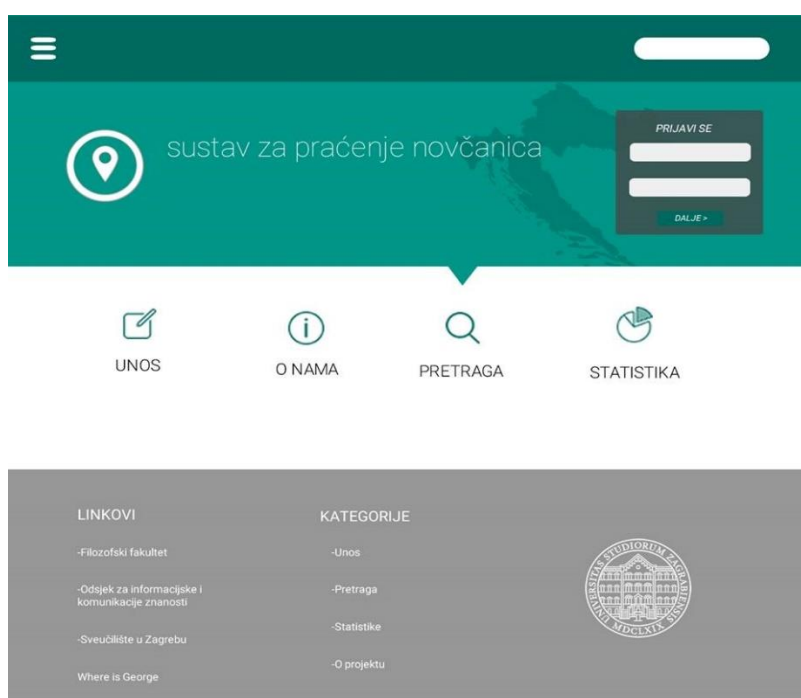
Bootstrap mreža razrađena je s nekoliko prijelomnih točaka (eng. *Break points*) za lakšu prilagodbu različitim dimenzijama uređaja što je također korišteno za jednostavniju i efikasniju izradu mobilne verzije (W3Schools.com, 2015.).

```
#novcanica > .flip3D > .front{  
  
    position: absolute;  
    -webkit-transform: perspective( 600px ) rotateY( 0deg );  
    transform: perspective( 600px ) rotateY( 0deg );  
    background:#FFF; width:400px; height:200px; border-radius: 7px;  
    -webkit-backface-visibility: hidden;  
    backface-visibility: hidden;  
    transition: -webkit-transform .5s linear 0s;  
    transition: transform .5s linear 0s  
  
}
```

Kodni blok 8: Primjer definiranja stilističkih atributa elementu pod nazivom klase 'front' koji se nalazi unutar elementa klase 'flip3d' koji se nalazi unutar elementa čiji *id* atribut nosi naziv 'novcanica'



Slika 5: Dizajn mobilnog sučelja kodiran pomoću *Bootstrap* razvojnog okvira



Slika 6: Izgled grafičkog sučelja nove verzije u razvoju

4.6. Implementacija skriptnih klijentskih jezika, otvorenog koda i API-a

Klijentski skriptni jezik neizostavan je dio svake web stranice jer omogućava brzu i laganu primjenu u obavljanju nekih jednostavnijih kalkulacija. Ti jezici, u našem slučaju JavaScript i JQuery se ne preporučaju koristiti u situacijama gdje moramo procesuirati povjerljive podatke primjerice dohvaćanje podataka iz neke baze podataka. Budući da se kod programiran tim jezicima izvršava na računalu klijenta, klijent ima uvid u cjelokupni izvorni kod stranice i samim time može manipulirati povjerljivim podacima pa čak i sabotirati stranicu. Zbog navedenih razlika ti jezici u našoj aplikaciji korišteni su za kreiranje animacija, provjeru formulara, ispisivanja poruka, dinamičnu promjenu sadržaja, implementaciju API-a sve u korist poboljšavanja korisničkog iskustva.

Neke od navedenih komponenti nisu izvorno napisani od naše strane već su preuzeti kao već spomenuti *otvoreni kod* koji se modificirao za naše potrebe i implementirao u kod web stranice. Iako se radi o otvorenom, besplatnom, javnom kodu koji je od naše strane modificiran nije moralno prepisati zaslugu. Tu se u igru uključuje pojam *informacijske etičnosti* koji u našem kontekstu govori da je poželjno naglasiti autora pripadajućeg otvorenog koda. Također potiče se ustupanje vlastitih programskih rješenja u obliku otvorenog koda ostatku zajednice kako bi se ostvario što veći kolektivni napredak.

Ovo su primjeri korištenja otvorenog koda i API-a u našem radu:

- Google Maps API, maplace.js (Kodni blok 9)
 - Korišten za iscrtavanje karte i ruta kretanja novčanica unutar karte

```

<script src="http://maps.google.com/maps/api/js?sensor=false"></script>
<script src="js/maplace.js"></script>
<script>
new Maplace({

    locations: Lokacije,
    map_div: '#gmap',
    generate_controls: false,
    show_markers: false,
    type: 'directions',
    draggable: false,
    directions_panel: '#route',
    afterRoute: function(distance) {
        $('#km').text(': '+(distance/1000)+'km');
    }).Load();

</script>

```

Kodni blok 9: Primjer korištenja maplace.js i Google Maps API koda za generiranje rute unutar Google potukaza

- Alterify.js
 - Korišten za personalizirano korištenje alert() JavaScript funkcije za ispisivanje poruka u novom prozoru (Kodni blok 10)

```

<script src="js/alertify.min.js"></script>
<script>alertify.error("Neuspješan unos. Odabrali ste krivi apoen.");</script>

```

Kodni blok 10: Primjer korištenja alterify.js koda za prikazivanje prilagođene poruke koja signalizira pogrešku

- OCRAD.js
 - Korišten za optičko prepoznavanje znakova novčanice (Kodni blok 11)

```

<script src="js/vendor/ocrad.min.js"></script>

<script>
    var image = new Image();
    image.src='slika.png'
    var canvas = document.getElementById('c1');
    canvas.width = image.naturalWidth;
    canvas.height = image.naturalHeight;
    var context = canvas.getContext('2d');
    context.drawImage(image, 0, 0);
    var parsiraniTekst = OCRAD(canvas);
</script>

```

Kodni blok 11: Primjer pozivanja OCRAD funkcije za optičko prepoznavanje teksta sa sadržaja prikazanog u <canvas> elementu

- Geoloc.js
 - Korišten za dohvaćanje i pretvorbu podataka o lokaciji korisnika (Kodni blok 12)

```

<script src="geoloc.js"></script>
<script>

    function pronadjiLokaciju(pozicija)

        {printAdress(pozicija.coords.latitude,
                     pozicija.coords.longitude);}

</script>

```

Kodni blok 12: Primjer pozivanja funkcije printAdress unutar geoloc.js datoteke za dohvaćanje podataka o lokaciji korisnika

4.7. Programiranje mrežne aplikacije

Za procesuiranje podataka o registriranim novčanicama, njihovu detaljnu obradu, verifikaciju, klasifikaciju i pohranu u bazu podataka koristili smo poslužiteljski programski jezik PHP. S ovim jezikom programirali smo jezgru i funkcionalnost naše web aplikacije.

Budući da se kod napisan ovim jezikom izvršava na poslužitelju (serveru), on nije dostupan na uvid korisniku već samo njegovi „izlazni“ podatci napisani unutar funkcije *echo* (Kodni blok 16). Tako sa ovim jezikom možemo manipulirati povjerljivim podacima iz određene baze podataka te nakon obrade ispisati rezultat u HTML kodu koji će biti vidljiv korisniku (The PHP group, 2015.).

Osim same obrade i unosa podataka u bazu podataka, ovaj programski jezik koristili smo za dohvaćanje podataka iz baze podataka (Kodni blok 15) te na temelju izvršavanja posebno definiranih upita prezentirali statistike vezane za kretanje registriranih novčanica.

Također jezik je korišten za slanje podataka među stranicama naše domene (Kodni blok 14) koje su pohranjene u sesijske varijable (eng. *sessions*) koje se uništavaju pri napuštanju domene ili ako je to definirano kodom (The PHP group, 2015.).

Tu prednost ovog programskog jezika iskoristili smo za mogućnost kreiranja i korištenja korisničkih računa kao i za slanje podataka o uspješnom ili neuspješnom unosu, prijavi, registraciji na temelju kojih smo pozivali određene funkcije u JavaScript programskom jeziku za kreiranje obavijesnih poruka. Važno je napomenuti da se spomenutom *echo* funkcijom može izvršiti bilo koji HTML kod što uključuje i izvršavanje naredbi skriptnih jezika čime se postiže vrlo važna interakcija između ta dva jezika. No ova „suradnja“ nije moguća samo u jednom smjeru. Moguće je podatke dobivene izvršavanjem skriptnog koda unutar *echo* naredbe iskoristiti u daljnjem dijelu PHP koda tako da podatke pretvorimo u JSON (*JavaScript Object Notation*) oblik ugrađenom funkcijom (Kodni blok 17). JSON struktura podatka je tekstualni zapis podatka ili strukture podataka neovisan o JavaScript kodu koji se može prenijeti na bilo koji drugi programski jezik (W3Schools, 2015.).

```

<?php

    session_start();
    header('Content-Type: text/html; charset=utf-8');
    $serijski = strtoupper($_POST["Serijski_broj"]);
    if(isset($_SESSION['username']))
    { $korisnik = $_SESSION['username'];}

    ...

    $_SESSION['stanje'] = 'uspjesno';

?>

```

Kodni blok 13: Primjer dohvaćanja poslanih podataka putem HTML obrasca i korištenje sesijskih varijabli

```

if ($_SESSION['stanje']=='krivi_unos')
{

    echo "<script>alertify.error('Neuspješan unos. Molimo Vas provjerite serijski broj i pokušajte ponovno.');

```

Kodni blok 14: Primjer korištenja *echo* funkcije za implementaciju HTML i skriptnog koda unutar PHP programskog koda

```

<?php

    header('Content-Type: text/html; charset=utf-8');
    $veza = mysqli_connect('localhost','root','', 'proba');
    mysqli_set_charset($veza, 'utf8');
    mysqli_query($veza, "SET NAMES utf8");

    $QUERY = "SELECT * FROM ime_tablice_korisnickih_racuna WHERE ID = $username";
    $rezultat = mysqli_query($veza, $QUERY);

?>

```

Kodni blok 15: Primjer koda korištenog za spajanje na bazu podataka i izvršavanje željenih SQL upita

```

<?php

    $data=file_GET_contents("http://maps.googleapis.com/maps/api/distancematrix/json?origins=$polaziste
&destinations=$cilj&language=en-EN&sensor=false");

    $data = json_decode($data);

    ...

?>

```

Kodni blok 16: Primjer korištenja ugrađene funkcije dekodiranje JSON strukture podataka

4.8. Optimizacija koda mrežne aplikacije

Prilikom programiranja poželjno je obratiti pažnju na optimizaciju i stilistiku programskog koda. Tako je prije konačnog puštanja aplikacije u digitalni svijet potrebno učiniti nekoliko koraka kako bi se stvorilo što bolje iskustvo budućim korisnicima i mogućim budućim programerima na stranici.

Neki od primjera optimizacije koda korištenih u našoj aplikaciji su:

- Pravilno formatiranje i komentiranje koda
- Uklanjanje mrtvog koda
- Skraćivanje koda korištenjem objektnog stila programiranja
 - Kod u kojem se određene programske radnje često izvršavaju na različitim entitetima poželjno je držati objektno orijentiranim. Sa tim postupkom definiramo klase i funkcije na temelju kojih ćemo stvarati objekte i izvršavati metode u svega jednoj liniji koda.
- Rangiranje provjere uvjeta temeljeno na vjerojatnosti
 - Ukoliko u kodu postoji niz kompleksnih provjera uvjeta, poželjno je rangirati uvjete po vjerojatnosti njihovog izvršavanja. Primjerice u našem slučaju, veća je vjerojatnost da će korisnik registrirati novčanicu od deset HRK u sustav nego onu od tisuću. Stoga novčanicu od deset kuna stavljamo kao prvu provjeru, a novčanicu od tisuću kao zadnju provjeru.
- Testiranje pregleda aplikacije na ekranima raznih rezolucija i poboljšavanje kompatibilnosti

- U našem radu bilo je važno da se aplikacija jednako prikazuje na svim uređajima i svim rezolucijama
- Uklanjanje pogrešaka u kodu temeljenih na dužem testiranju.

4.9. Pretvorba mobilne verzije mrežne aplikacije u mobilnu android aplikaciju

Budući da je naša mrežna aplikacija optimizirana za pregledavanje u internetskim preglednicima na mobilnim uređajima, odlučili smo aplikaciju konvertirati u *android* (.apk) aplikaciju. Programiranje iste aplikacije novim programskim jezikom oduzelo bi mnogo vremena i truda te smo se zbog toga odlučili iskoristiti besplatni open-source on-line alat za pretvorbu prilagodljivih mobilnih mrežnih stranica u mobilne aplikacije namijenjene popularnim mobilnim platformama (Android, iOS, Windows Phone) po imenu *PhoneGap* (Adobe Systems Inc, 2015.).

Ono što smo postigli ovom pretvorbom zapravo je stvaranje prečaca na mobilnom android uređaju koji mobilnu stranicu ne učitava u svom pregledniku već u posebnom *webview* načinu prikaza web stranice. Ovim pristupom sav sadržaj na web stranici biti će prikazan unutar zasebne aplikacije dok se sve eksterne poveznice otvaraju u pregledniku. (Apache 2.0, 2015.).

Ovaj alat od korisnika traži kreiranje korisničkog računa koji može biti besplatan ukoliko želimo osnovne mogućnosti koje su bile dovoljne u našem slučaju ili komercijalan ukoliko korisnik želi konvertirati veći broj aplikacija i koristiti dodatne usluge. Sve što je potrebno napraviti je poslati izvorni kod mrežne aplikacije, odrediti ime, opis, verziju te ikonu aplikacije. Nakon samo nekoliko minuta ovaj alat kreira aplikaciju (.apk) koju možemo preuzeti na svoj uređaj.

4.10. Sigurnost i zaštita podataka

Informacijska sigurnost postala je jako važan faktor pri kreiranju i očuvanju bilo kojeg informacijskog sustava. Sigurnost web aplikacija je ugroženija nego kod nekih drugih tipova aplikacija.

U današnje vrijeme popularni su razni napadi na web stranice od kojih su najčešći:

- Code injection
- DDOS napadi

Code injection napadi funkcioniraju na način da se kroz tekstualne forme HTML elemenata, čiji se sadržaj dalje procesira kroz PHP ili neki drugi programski jezik, umeće zlonamjerni programski kod kojim se želi u procesiranju podatka dovesti do nekog oblika eksploatacije podataka.

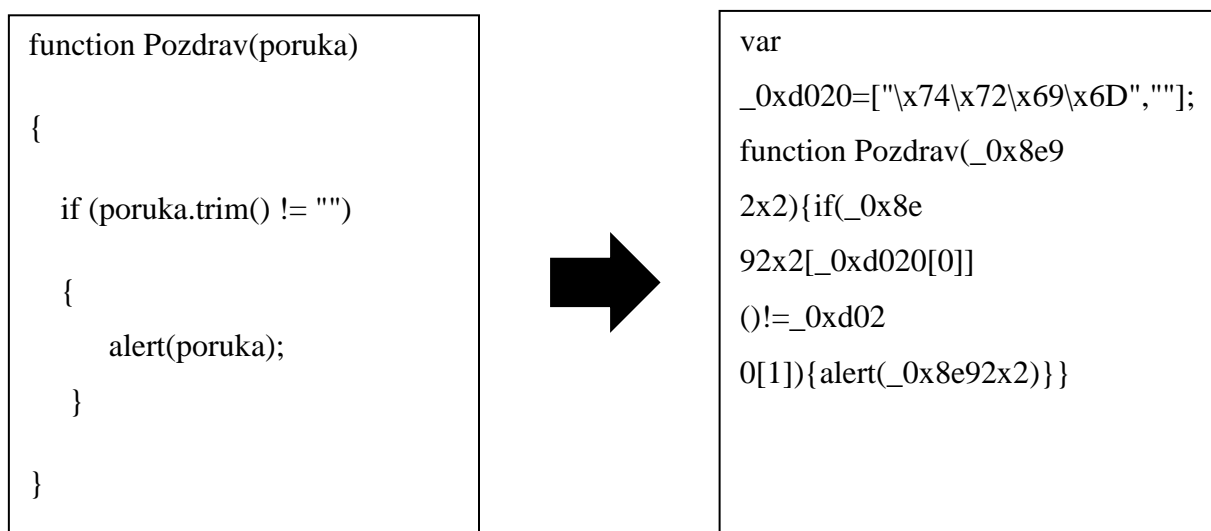
Najčešći *code injection* napad je *SQL injection* napad kada se u formu umeće određeni blok SQL koda kojim se može manipulirati bazom podataka. SQL injection metoda se bazira na tome da se unutar tekstualne forme unose navodnici ("), a potom neka SQL naredba kako bi se unutar programskog koda „zatvorila“ predviđena naredba i „usadila“ željena zlonamjerna SQL naredba koju će programski jezik izvršiti (W3Schools.com, 2015.).

Kako bi se efikasno obranili od takve vrste napada na našoj stranici, koristili smo ugrađenu PHP funkciju `mysqli_real_escape_string()` koja kao argument prima SQL naredbu u obliku niza znakova string te umeće znak za izbjegavanje (eng. *escape characer* - „\“) prije znakova (NUL (ASCII 0), \n, \r, \, ', ", i Control-Z) kako bi se izbjeglo njihovo parsiranje (The PHP group, 2015.).

Također pri spremanju nekih privatnih podataka u bazu podataka poput lozinke korisnika (koju nikada nećemo prezentirati) šifrirali smo MD5 hash algoritmom. Ono što MD5 hash algoritam radi je da šifrira tekst *trap door* funkcijom. *Trap door* funkcija oblik je funkcije koji se lako računa u jednom smjeru a vrlo teško i sporo u suprotnom smjeru. Tako da je algoritmu za hashiranje potrebno svega nekoliko stotinki da šifrira prosječnu riječ od 8 znakova, no potrebno je jako dugo vremena da se hashirani heksadecimalni zapis duljine 32 znaka pretvori u prvobitni tekst. Ovim postupkom pri prijavljivanju na web stranicu uspoređuju se hash zapisi lozinke što u slučaju krađe podataka iz baze podataka može zaštititi korisnika od krađe privatnih podataka i manipulacije korisničkim računom. Za ovakvu vrstu zaštite također smo koristili ugrađenu PHP funkciju `md5()` koja kao argument prima niz znakova te vraća hashirani niz znakova (PHP.net, 2015.). Primjer rezultata hash funkcije za niz znakova „pozdrav“ izgleda ovako: „pozdrav“ → „0a2518226b05e4b72d8ffbe7e992704“.

Cross-site scripting je vrsta *code injection* napada kojim se u izvorni kod web stranice umeće klijentski skriptni jezik poput Javascripta ili JQuery-a s ciljem eksploatiranja podataka ili otkrivanja sigurnosnih propusta u nekim složenijim aplikacijama baziranim na klijentskom skriptnom jeziku. Ovakva vrsta napada vrlo je popularna u web igricama s ciljem varanja i manipuliranja rezultatom (Wikipedia, 2015.).

Kako bi se zaštitili od ovakve vrste napada sav JavaScript i JQuery kod na web stranici je „pomučen“, odnosno na neki način kodiran tako da su uklonjeni svi elementi prirodnog jezika, segmentacije koda te je umetnut „mrtvi kod“ kako bi se čovjeku otežalo razumijevanje koda (npr. Kodni blok 18). Alat korišten pri „zamućivanju“ koda naziva se *Free Javascript Obfuscator*. Ovim postupkom otežavaju se mogući *Cross-site scripting* napadi ili krađa JavaScript koda (Javascriptobfuscator.com, 2015.).



Kodni blok 17: Primjer zamućivanja JavaScript koda

4.11. Budućnost web aplikacije

Nakon završetka finalne verzije naše web aplikacije, preostaje nam na neki način potaknuti njeno korištenje. Osnovni plan je promovirati aplikaciju putem društvenih mreža i elektroničke pošte. Nakon određenog vremena nadamo se da će se prikupiti dovoljno podataka kako bi se mogli ostvarivati pogodci i postići svrha mrežne stranice. Ukoliko se ova ideja pokaže kao uspješna i zanimljiva korisnicima, u obzir dolazi dodatno usavršavanje i optimizacija aplikacije kao i kvalitetnija promidžba.

Budući da se u ovoj fazi projekta ne razmišlja o bilo kakvom načinu zarade, kao vlasnicima prikupljenih podataka u budućnosti javlja se opcija prodaje podataka interesnim skupima jer se korisnicima omogućava samo uvid u mali dio podataka. U ovom dijelu dotakao bih se već spomenutog pojma Velikih podataka koji je postao važan faktor u poslovnom odlučivanju. Tako se javlja i nada da će se podatci u velikoj količini prikupljeni kroz naredne godine

poslužiti kao potpora u poslovnom odlučivanju ili, kao u spomenutom slučaju *Where's George* mrežne stranice, iskoristiti u istraživačke svrhe.

5. Zaključak

U ovom sam radu opisao proces kreiranja informacijskog sustava u obliku web stranice i mobilne aplikacije od početne ideje do završnog proizvoda i planova za budućnost.

Budući da se ovaj rad temelji na konkretnom primjeru detaljno sam obradio svaki korak vezan za nastajanje web aplikacije. Krenuvši od same ideje za kreiranjem jedne web aplikacije objasnio sam podrijetlo nastanka ideje, naveo i usporedio srodne web stranice i proučavanjem istih uočio potencijalne mane i nedostatke kako bi se na stranici postiglo što bolje korisničko iskustvo.

Nakon određivanja smjernica za kreiranje aplikacije pažnju sam prebacio na njenu tehničku izvedbu. Započevši s osiguravanjem internetske domene i uvjeta za prikazom naše web aplikacije na internetu, naglasak sam potom stavio na definiranje nekih logističkih okvira po kojoj naša aplikacija mora funkcionirati (određivanje informacijske sheme i ustroja baze podataka). U nastavku sam na konkretnim primjerima objasnio primjenu HTML i CSS jezika za označivanje u kreiranju strukture i dizajna web aplikacije te njenu prilagodbu u *bootstrap* razvojnom okviru za prikazivanje sadržaja na ekranima mobilnih uređaja. Nakon strukturiranja i dizajniranja web stranice na konkretnim primjerima pokazao sam programska rješenja kojima smo dodali funkcionalnost našoj web aplikaciji. Prikazao sam primjene skriptnih klijentskih jezika kao i uporabu poslužiteljskog jezika za obavljanje složenijih programskih potreba. Na primjerima sam objasnio njihove razlike i mogućnosti, te situacije u kojima je poželjno koristiti određeni programski jezik. Na posljatku sam se dotakao važnosti očuvanja sigurnosti podataka te u konačnici pretvorbe mobilne web stranice u mobilnu aplikaciju za operacijski sustav android.

Mislim da sam ovim radom uspio obuhvatiti sve potrebne elemente u kreiranju moderne web aplikacije. Veliki naglasak stavio sam na uporabu otvorenog koda, implementaciji gotovih programskih i dizajnerskih rješenja te informatičkoj etičnosti koji danas uvelike olakšavaju i ubrzavaju proces kreiranja web stranica i programskih rješenja generalno.

6. Literatura

1. Apache 2.0, „**Web view**“, <http://developer.android.com/reference/android/webkit/WebView.html>, (posjećeno 12.9. 2015.), 2015.
2. Adobe Systems Inc, „**PhoneGap**“, <http://phonegap.com>, posjećeno(10.9.2015.), 2015.
3. Association Européenne des EuroBillTrackers, „**EuroBillTracker**“, <http://en.eurobilltracker.com>, (posjećeno 30.8.2015.), 2015.
4. Begušić, D., „**Inžinjerska grafika**“, Split, 2011.
5. CARNet, „**Društveni softver: Web 2.0**“, <http://www.carnet.hr/tematski/drustvenisoftver/pojmovnik>, (posjećeno 27.8.2015.), 2009.
6. Dan Durrans, „**Statistics**“, <http://www.findlizzy.co.uk/statistics>, (posjećeno 26.8.2015.), 2015.
7. Donald, G. McNEIL Jr., „**Predicting Flu With the Aid of (George) Washington**“, http://www.nytimes.com/2009/05/04/health/04model.html?_r=2&scp=2&sq=wheresgeorge&st=cse, (posjećeno 30.9.2015.), 2009.
8. EuroTracer, „**Statistics**“, <http://www.eurotracer.net/statistics>, (posjećeno 30.8.2015.), 2015.
9. Javascript Obfuscator, „**JavaScript Obfuscator**“, <https://javascriptobfuscator.com>, posjećeno(2.9. 2015), 2015.
10. Kocijan,K., „**Big Data: kako smo došli do velikih podataka i kamo nas oni vode**“ u: „**Komunikacijski obrasci i informacijska znanost**“, Zavod za informacijske studije, Zagreb, 2014, str. 37-62.
11. Preston, G., **Kako funkcionira Internet**, Algoritam d.o.o, Zagreb,2004.
12. Radovan, M., „**Baza podataka : Relacijski pristup i SQL**“, Informator, Zagreb, 1993.
13. Radovan, M. , „**Projektiranje informacijskih sustava**“, Informator, Zagreb, 1991.
14. ScienceBlog.com, „**Web game provides breakthrough in predicting spread of epidemics**“, <http://scienceblog.com/9874/web-game-provides-breakthrough-in->

- predicting-spread-ofepidemics/#OoCuBWwUVUbqPegk.97, (posjećeno 24.8.2015.), 2006.
15. The PHP group, „**PHP Manual**“, <http://php.net/manual/en>, (posjećeno 16.9.2015.), 2015.
 16. TrackDollar.com, „**TrackDollar.com**“, <http://trackdollar.com>, (posjećeno 29.8.2015.), 2015.
 17. W3Schools, „**W3Schools**“, <http://www.w3schools.com>, (posjećeno: 16.9.2015.)
 18. Where's George? L.L.C, „**Where's George?**“, <http://www.wheresgeorge.com>, (posjećeno 24.8.2015), 2015.
 19. Where's George? L.L.C., „**Where's Willy?**“, <http://www.whereswilly.com>, (posjećeno 24.8.2015), 2015.
 20. Wikipedia, „**Wikipedia**“, <https://en.wikipedia.org>, (posjećeno 29.8.2015.), 2015.
 21. www.cda-money.ca, „**Candian Money Tracker**“, <http://www.cdn-money.ca/cdnmoney/servlet/cmtMain>, (posjećeno 30.8.2015.), 2015.